

REMARKS

In the pending application, claims 1-10 were presented for examination. Applicants have canceled claim 10, amended claims 1-9 and have added new claims 11-14. Claims 1-9, and 11-14 are currently pending in the present application. Applicants respectfully request reconsideration of the pending claims in view of the following statements and/or amendments to the claims.

The specification was objected to because of missing information on pages 1, 5, 6, and 28 relating to related patent applications. Applicants have attached replacement pages 1, 5, 6, and 28 containing the requested information.

Claims 1-5, 7-10 were rejected under 35 U.S.C. 103(a) based on Helland et al. (U.S. Patent No. 6,134,594) and Hardiman et al. (U.S. Patent No. 5,504,672).

Claim 1, as amended, is directed to a system for developing a software application for manipulating data associated with an asset. Claim 1 recites:

at least one processing unit;

at least one memory store operatively connected to the processing unit;

extensible N-tier software resident in and executable within the at least one processing unit;

an inventory of software components resident in the memory store wherein a plurality of tiers are generated from the inventory of software components using the N-tier software, each tier being associated with at least one other tier, and each tier comprising a plurality of software components and performing a predetermined function relating to an asset, each software component comprising a software object;

an input device operatively in communication with the processing unit;
and

an output device operatively in communication with the processing unit.

Referring to Helland et al., the reference is directed to a server architecture. Helland et al., however, provides no teaching of: "an inventory of software components resident in the memory store wherein a plurality of tiers are generated from the inventory of software components using the N-tier software, each tier being associated with at least one other tier, and each tier comprising a plurality of software components and performing a predetermined function relating to an asset, each software component comprising a software object", as recited in independent claims 1 and 10, as amended. Further, Hardiman et al. does not provide any teaching of the foregoing claim limitations.

Accordingly, because neither Helland et al. nor Hardiman et al. teach all of a limitations of the claims 1-5, 7-9, as amended, applicants submit that the claims are allowable over these references.

Claim 6 is rejected under 35 U.S.C. 103(a) based on Helland et al. and Hardiman et al. and McDonald et al. (U.S. Patent No. 5,978,582). As discussed above, Helland et al. and Hardiman et al. do not teach, among other limitations, "an inventory of software components resident in the memory store wherein a plurality of tiers are generated from the inventory of software components using the N-tier software, each tier being associated with at least one other tier, and each tier comprising a plurality of software components and performing a predetermined function relating to an asset, each software component comprising a software object", as recited in claim 1, from which claim 6 depends. Further, McDonald et al. does not teach the foregoing limitations.

Accordingly, because the proposed combination of Helland et al., Hardiman et al. and McDonald et al. does not teach all of the limitations of the claim 6, applicants submit that claim 6 is allowable over these references.

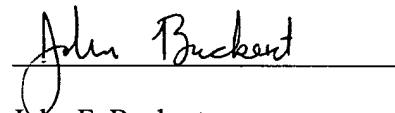
Applicants have added claims 11-14 to claim particular aspects of the

present invention. Support for these claims can be found within the specification. Applicants submit that no new subject matter has been added by claims 11-14. Claims 11-14 are believed to be allowable for at least the same reasons recited above with respect to claims 1-5, and 7-9.

If for any reason the Examiner feels that consultation with applicants' attorney would be helpful in the advancement of the prosecution, the Examiner is invited to call applicants' attorney at the telephone number listed below.

If there are any charges due with respect to this Amendment or otherwise, please charge them to Deposit Account No. 06-1130.

Respectfully submitted,

A handwritten signature in cursive script, reading "John F. Buckert", is written over a horizontal line.

John F. Buckert
Reg. No. 44,572
Attorney For Applicant

Date: 5/12/04

**METHOD AND SYSTEM AND ARTICLE OF MANUFACTURE FOR AN N-TIER
SOFTWARE COMPONENT ARCHITECTURE OILFIELD MODEL**

CROSS-REFERENCES TO RELATED APPLICATIONS

This nonprovisional U.S. national application, filed under 35 U.S.C. § 111(a), claims, under 37 C.F.R. § 1.78(a)(3), the benefit of the filing date of provisional U.S. national application no. 60/173,914, attorney docket no. D5407-00109, filed on 12/29/99 under 35 U.S.C. § 111(b), the entirety of which is incorporated herein by reference. It has been proposed in co-pending U.S. patent application Ser. No. 09/746,155, attorney docket no. D5407-00109, filed on December 22, 2000 with a U.S. Express Mail number of EL675616662US and incorporated herein by reference, to provide a system and method for software design of software architectures and, in particular, to provide for the design of a software component architecture for the development of extensible tier software component applications such as is used herein. It has been further proposed in co-pending U.S. patent application Ser. No. 09/746,362, attorney docket no. D5407-00123, filed on December 22, 2000 with a U.S. Express Mail number of EL675616645US and incorporated herein by reference, to provide a system and method for development of software applications using software component architecture for the development of extensible n-tier software applications.

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to software applications having an extensible N-tier design and architecture and, in particular, to a new and improved method of software application creation for

Invention: METHOD OF AND SYSTEM AND ARTICLE OF MANUFACTURE
FOR AN N-TIER SOFTWARE COMPONENT ARCHITECTURE OILFIELD MODEL

Inventors: Green, David et al.

Fig. 27 is a representation of a further exemplary display from the exemplary application;

Fig. 28 is a block diagram of the processing occurring when adding a new well to the field in the exemplary application.

DETAILED DESCRIPTION

5 Referring generally to Fig.1 through Fig.6, the present invention comprises a methodology that applies an engineering and manufacturing oriented approach to software production based on a well-defined architecture. As used herein, “manufacturing” implies a method analogous to a software factory. Using the present invention methodology, software application development can proceed as if it was a software manufacturing process with an
10 assembly line capable of assembling all types of intellectual property quickly and at the lowest cost.

It has been proposed in co-pending U.S. patent application Ser. No. 09/746,155 attorney docket no. D5407-00109, filed on December 22, 2000 with a U.S. Express Mail number of EL675616662US and incorporated herein by reference, to provide a system and method for
15 software design of software architectures and, in particular, to provide for the design of a software component architecture for the development of extensible tier software component applications such as is used herein. Utilizing that system and method, it has been further proposed in co-pending U.S. patent application Ser. No. 09/746,362, attorney docket no. D5407-00123, filed on December 22, 2000 with a U.S. Express Mail number of EL675616645US and
20 incorporated herein by reference, to provide a system and method for development of software applications using the software component architecture for the development of extensible n-tier software applications.

The present invention uses an “N-tier architecture” paradigm as described in co-pending U.S. patent application Ser. No. 09/746,155, and co-pending U.S. patent application Ser. No. 09/746,362. In an N-tier architecture, all functionality is broken down at the system level into logical chunks or tiers 30 that perform a well-defined business function. In the present invention’s N-tier architecture there is no limit to the number of tiers 30.

The present invention’s N-tier software design architecture is employed to develop software components 20. As those of ordinary skill in the programming arts will appreciate, “software components” are language independent and may be implemented in any of a number of computer languages including without limitation FORTRAN, C, C++, JAVA, assembler, or the like or any combination thereof. As those of ordinary skill in the programming arts will appreciate, “N-tier” in the prior art may be thought of as implying a hierarchy such as with protocol stacks. However, as used herein, “N-tier” describes an architecture that is characterized by a plurality of “N” tiers 30, each of which has a specified type and a specified interface. Although a hierarchy can be defined for the tiers 30, no one hierarchy is mandatory in the N-tier architecture of the present invention.

Each software component 20 to be developed is associated with at least one tier 30, depending upon the nature of the functions to be performed by that software component 20 and tier 30. The present invention specifies a method and a system for using architectures to implement a N-tier system wherein a software component designer can design or select each software component 20 to perform specified functionality and ensure that each software component 20 has the interfaces specified by the architecture for that tier 30.

user, e.g. a user seated at a keyboard, view 7002 searches the appropriate data repositories, e.g. an SQL database and/or a DLIS tape, and returns the query results to the user's screen 15 for viewing.

In the preferred embodiment, Visual tier 30 allows a user to see details of any business
 5 model 31, such as the one in Fig. 24. In this example, access to model 31 is limited by controller 7003 in Visual tier 30 which uses a model-view-controller paradigm. Access via controller 7003 may be configurable, i.e. controller 7003 may be read-only or any other permissions based on programmatic control. For example, in an alternative embodiment, controller 7003 allows a user to edit business model 31 as well as view it. This type of functionality allows shipment of the
 10 same application to customers with multiple needs, resource levels, or levels of security. For example, a customer that only needs a model viewer purchases the application with controller 7003 in a read-only mode, whereas a customer that wants to edit models 31 as well as view them purchases the application with a controller 7003 capable of both viewing and editing. A customer could even purchase the application with both controllers 7003, allowing some users access to
 15 the read-only controller 7003 while others would have access to the view and edit controller 7003.

In the operation of a preferred embodiment, referring still to Fig. 24, requirements for an application for oil field 201 are first determined in accordance with the method of building an N-tier application of the present invention and co-pending U.S. patent application Ser. No.
 20 09/746,157. Software components 20 obtained, created, and/or purchased to implement the functionality required to satisfy the requirements for oil field 201. These software components 20 are then associated with tiers 30, and an application generated to implement the functionality required to satisfy the requirements for oil field 201.